each bit defines an error correcting capability based on whether the bit is set to 1 or 0. The meaning of each bit is provided in Table 6.

Table 6	
Byte Bit Position	**Meaning**
Bit 0	Other
Bit 1	Unknown
Bit 2	None
Bit 3	Single Bit Error Correcting
Bit 4	Double Bit Error Correcting
Bit 5	Error Scrubbing

```
dw    BIT_FIELD_ID                    ; Data identifier
db    "Error Correcting Capability",0 ; Field description
db    1                               ; Bytes used for bit-field
db    "No",0                          ; String to use when bit = 0
db    "Yes",0                         ; String to use when bit = 1
; Begin bit field encoding for this value
db    00h, "Other",0
db    01h, "Unknown",0
db    02h, "None",0
db    03h, "Single Bit Error Correcting",0
db    04h, "Double Bit Error Correcting",0
db    05h, "Error Scrubbing",0
dw    ENDOF_BIT_FIELD_ID
```

As illustrated, the remaining two bits of the bit field are not defined, and as such, the ENDOF_BIT_FIELD_ID key is used to end the structure definition, such that the remaining two bits do have to be defined in the template file.

D. ENUM_ID and ENDOF_ENUM_ID

This key defines an enumerated value, meaning that the field's numerical value represents some type of defined setting. Enumerated values are always one byte.

Encoded Value:

ENUM_ID	0xFFF4
ENDOF_ENUM_ID	0xFFF5

Format:

```
dw      ENUM_ID
db      Null terminated string for field title
db      xx – Enumerated value setting
db      Null terminated string describing the previous enumerated value
...repeat this section until all values are defined...
dw      ENDOF_ENUM_ID
```

Notes:

The ENDOF_ENUM_ID key indicates the end of the enumerated values so that undefined values do not have to be defined in the template file. For example, an enumerated value stored in the SMBIOS database is on byte long or 8 bits. That means that there are 2^8 possible values. If less than all values are currently defined, the ENDOF_ENUM_ID may be used after the last defined value.

Assembler Data Example:

Illustrated below is the structure definition for data related to the processor type, which is Type 4, offset 05h. The enumerated value for the offset has 6 possible values, with each value of the enumerated value indicating a different type of processor. Table 7 provides the processor designation for each byte value of the enumerate value.

Table 7

Byte Value	Meaning
01h	Other
02h	Unknown
03h	Central Processor
04h	Math Processor
05h	DSP Processor
06h	Video Processor

```
dw      ENUM_ID                 ; Data identifier
db      "Processor Type",0      ; Field Description
db      01h, "Other",0
db      02h, "Unknown",0
db      03h, "Central Processor",0
db      04h, "Math Processor",0
db      05h, "DSP Processor",0
db      06h, "Video Processor",0
dw      ENDOF_ENUM_ID
```

E. GROUPED_BIT_FIELD_ID, GBF_END_GROUP, and ENDOF_GROUPED_BIT_FIELD_ID

These descriptor keys define a grouped bit field. A grouped bit field differs from a normal bit field in that the groups of bits collectively represent some sort of setting of the computing system similar to an enumerated value. For example, a grouped bit field having a size of 1 byte may have bits 0-2 representing 8 different settings based on their value, while bits 3 and 4 represent 4 other settings, etc. In this instance, the GROUPED_BIT_FIELD_ID descriptor key represents the beginning of the grouped bit field. The GBF_END_GROUP descriptor key indicates the end of the definitions for each group within the grouped bit field. Similar to an enumerated value, each group within the grouped bit field contains description strings for each possible value of the bit group. Finally, since the number of possible values per group is defined as 2^n (where n is the number of bits in the group), all possible values may not be defined. As such, the ENDOF_GROUPED_BIT_FIELD_ID indicates the end of the grouped bit definitions, so that undefined groups of bits do not have to be defined in the template file. For example, if the byte size is 1, but only one bit group from bits 3:0 is defined, the remaining bit groups do not have to be defined in the template file.

Encoded Value:

GROUPED_BIT_FIELD_ID	0xFFF6
GBF_END_GROUP	0xFFF7
ENDOF_GROUPED_BIT_FIELD_ID	0xFFF8

Format

```
dw    GROUPED_BIT_FIELD_ID
db    Null terminated string for field title
db    Number of bytes occupied by the grouped bit field
(Begin bit field group encoding)
db    Group start bit
db    Group end bit
db    Null terminated string of group description
db    Bit group value
db    Null terminated string describing the above value
. . . repeat this section for each possible value in the group . . .
db    GBF_END_GROUP
. . . repeat this section for all groups . . .
dw    ENDOF_GROUPED_BIT_FIELD_ID
```